Hidden Markov Model for Control Strategy Learning

Jie Yang, Yangsheng Xu

CMU-RI-TR-94-11

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

May 1994

Contents

List of Figures

Abstract

This report presents a method for learning a control strategy using the hidden Markov model (HMM), i.e., developing a feedback controller based on HMMs. The HMM is a parametric model for non-stationary pattern recognition and is feasible to characterize a doubly stochastic process involving observable actions and a hidden decision pattern. The control strategy is encoded by HMMs through a training process. The trained models are then employed to control the system. The proposed method has been investigated by simulations of a linear system and an inverted pendulum system. The HMM-based controller provides a novel way to learn control strategy and to model the human decision making process.

1 Introduction

An intelligent controller has the ability to comprehend, reason, and learn about processes and environment. Because an intelligent control system is complex, analytic method in control theory is insufficient and inefficient for analyzing and designing an intelligent control system. Various methods have been proposed for designing intelligent control systems such as pattern recognition method [1, 2], fuzzy control [3, 4], and neurocontrol [5, 6, 7].

A controller maps its input onto an appropriate set of control actions. The correspondence between pattern recognition and control can be considered as the learned response of the control system to known patterns in the input data. The power of fuzzy sets and neural networks lies in their ability to represent the mapping of a controller. It has been proved that any continuous nonlinear mapping can be approximated as exactly as needed with a finite set of fuzzy variables, values, and rules [8]. In a typical neural network learning application, the desired mapping is static. The hidden assumption is that the nonlinear static map generated by the neural network can adequately represent the system's behavior [9]. Some mappings in the control systems, however, are non-stationary. Human performance is an example. Human performance is the actions and/or reactions of humans under specified circumstances. Actions reflect human skill of performing a certain task and reactions reflect the control strategy to environment. A human associates responses with stimuli, actions with scenarios, labels with patterns, and effects with causes. Because both human decision and sensory processes are stochastic, human control actions differ even when inputs are the same. In short, the human control strategy is non-linear, non-deterministic, and non-stationary, and it is necessary to model such a mapping with an appropriate tool.

The hidden Markov model (HMM) is a powerful tool for pattern recognition of the non-stationary stochastic process [10]. HMM is a doubly stochastic process: the hidden underlying stochastic process can only be observed through another set of stochastic processes. In addition, the HMM is a parametric model that can be optimized with efficient algorithms. HMMs has been successfully used in speech recognition [10, 11, 12, 13]. Recently their effectiveness has been studied in force analysis, task-context final segmentation, and extraction of discrete finite-state Markov signals [14, 15, 16]. We have successfully applied HMMs to human action modeling [17, 18]. Action and reaction are two important aspects of human performance. Both of them play important roles in human performance. For example, playing tennis is a complicated process involving actions and reactions. A tennis player determines his strategy based on his experience and the immediate information, such as ball's incoming speed and height. The player's actions will directly influence his reactions; hence, high-quality performance requires both good reactions and good actions. In this report, we discuss the problem of reaction modeling and application of HMMs to control strategy learning.

The objective of a learning controller is to acquire a control strategy from experience to achieve certain desired goals. In general, an HMM-based controller can be regarded as a means of learning a control strategy from a teacher. The control strategy modeled with HMMs is used for controlling the system. For a given system, the control strategy is partitioned into certain decision patterns, and these patterns are described by corresponding

HMMs. During the training process, HMMs learn a control strategy by adjusting their parameters. During controlling the system, at each "sampling time," the controller evaluates the feedback signal using the trained HMMs, and their scores (probabilities) are used to generate the controller output, i.e., the patterns with higher probabilities contribute more to the controller output.

2 Hidden Markov Modeling

2.1 Hidden Markov Model

A hidden Markov model is a collection of finite states connected by transitions. Each state is characterized by two sets of probabilities: a transition probability, and either a discrete output probability distribution or a continuous output probability density function which, given the state, defines the condition probability of emitting each output symbol from a finite alphabet or a continuous random vector.

An HMM can be defined by:

- A set of states {S}, with an initial state S_I and a final state S_F
- The transition probability matrix, $A = \{a_{ij}\}$, where a_{ij} is the transition probability of taking the transition from state i to state j
- The output probability matrix B. For a discrete HMM, $B = \{b_j(O_k)\}$, where O_k represents a discrete observation symbol. For a continuous HMM, $B = \{b_j(\mathrm{x})\}$, where x represents continuous observations of k-dimensional random vectors

In this report, we consider only a discrete HMM. For a discrete HMM, a_{ij} and $b_j(O_k)$ have the following properties:

$$a_{ij} \geq 0, \;\; b_j(O_k) \geq 0, \;\; \forall\, i,j,k, \tag{1}$$

$$\sum_j a_{ij} = 1 \;\; \forall\, i, \tag{2}$$

$$\sum_k b_j(O_k) = 1, \;\; \forall\, j. \tag{3}$$

If the initial state distribution $\pi = \{\pi_i\}$, the complete parameter set of the HMM can be expressed compactly as

$$\lambda = (A, B, \pi). \tag{4}$$

For a detailed description of the theory and computing HMM, the readers are referred to [10, 12].

2.2 Concept and Approach

Pattern recognition can be used for classifying objects or processes of unknown origin into predetermined classes. The problem of pattern recognition is usually characterized by a description and classification of a set of objects or processes. Many control problems can be described as pattern recognition problems. For example, in an on-off room-temperature control system, the task of a controller is to maintain the room temperature based on two patterns. However, controller usually have to identify infinite number of patterns and their outputs are continuous. Therefore, more complex techniques are required for controlling such systems based on a pattern recognition approach.

In a feedback system, the controller input is the error signal and the output is the control signal. The feedback control strategy, in general, is a function of the error signal, its history, and the control signal history. Usually, the feedback and control signals are multidimensional. If the control strategy depends only on the finite history of the feedback signal, the critical issue is to determine decision patterns, i.e., the number of HMMs for characterizing the decision patterns, because of no fixed pattern available for a controller. In order to obtain finite decision patterns, we can partition control space into finite patterns, model these patterns, and then generate control signals based on these models. We have developed the following technique to implement an HMM-based controller.

1. Record the controller input and output data, which are the feedback signals $e(i)$ and control signals $u(i)$. These data will be used for training HMMs. Assuming $u(i)$ depends only on k samplings of $e(i)$, the correspondence between $u(i)$ and the feedback signals can be established.

2. Partition control patterns and quantize the feedback signals into finite levels.
   (a) Partition $[Umin,\ \ Umax]$ into N patterns:

$$U = \{U_1, U_2, \ldots, U_N\}. \tag{5}$$

   (b) At each time $i = 1, 2, \ldots, M$, $u(i)$ belongs to one of the patterns, U_j, and corresponds to a set of sequences $\{E(i)\}$, where

$$\{E(i)\} = \{e(i), e(i-1), e(i-2), \ldots, e(i-k)\}, \quad k > 0. \tag{6}$$

3. Handmark each $\{E(i)\}$ to corresponding one of $\{U_j\}$, where $i = 1, 2, \cdots, M$ and $j = 1, 2, \cdots, N$.

4. Use HMM to describe each U_j, $j = 1, 2, \cdots, N$, and train the models by the data.

5. Put the trained models into the model bank for controlling the system. At each sampling time, $\{E(i)\}$ is scored by all trained models for obtaining the probabilities of decision patterns U_j, $j = 1, 2, \ldots, N$, matching with $\{E(i)\}$:

$$P(U_1), P(U_2), \ldots, P(U_N). \tag{7}$$

Then $P(U_j)$, $j = 1, 2, \ldots, N$, are sorted in a decreasing order and the first m $P(U'_j)$ are employed as weights to compute the controller output:

$$u = \frac{\sum_{j=1}^{m} P(U'_j) \cdot u'_j}{\sum_{j=1}^{m} P(U'_j)}, \tag{8}$$

where U'_j, $j = 1, 2, \ldots, m$, stand for the sorted patterns and u'_j stands for the corresponding control signal values.

In next section, we discuss how to develop an HMM-based controller in detail.

3 HMM-Based Controller

Figure 1 shows the basic configuration of an HMM-based controller which consists of four components: signal preprocessor, pattern evaluation unit, model bank, and control signal generator. The signal preprocessor measures the values of feedback signals, maps the range of values of measured signals onto corresponding universes of discourse, and converts the input data into suitable symbols which will be used by HMMs. The pattern evaluation unit estimates the probabilities that the input signals match with the models in the model bank. The model bank is the kernel of an HMM-based controller; it contains the trained HMMs which represent the most likely decision patterns of controller. The models in the model bank are trained by training examples. The control signal generator generates control signals based on the pattern evaluations, and converts the range of values of the control signals into corresponding universes of discourse.

3.1 Preprocessor

The measured feedback signals are preprocessed for appropriate enhancement. First they are filtered to eliminate noise. Then the resulting sample is converted into finite symbols because we use discrete HMM for describing decision patterns. In a multi-input system, the feedback signals are sequential vectors. Vector quantization (VQ) technique [19] is a suitable tool to map real vectors onto finite symbols. A vector quantizer is completely decided by a codebook which consists of a set of the fixed prototype vectors. A description of the VQ process includes: (1) the distortion measure, and (2) the generation of the certain number of prototype vectors. In the implementation, the squared error distortion measure is used, i.e.,

$$d(\mathrm{x}, \hat{\mathrm{x}}) = \|\mathrm{x}, \hat{\mathrm{x}}\| = \sum_{i=0}^{k-1} (x_i - \hat{x}_i)^2. \tag{9}$$

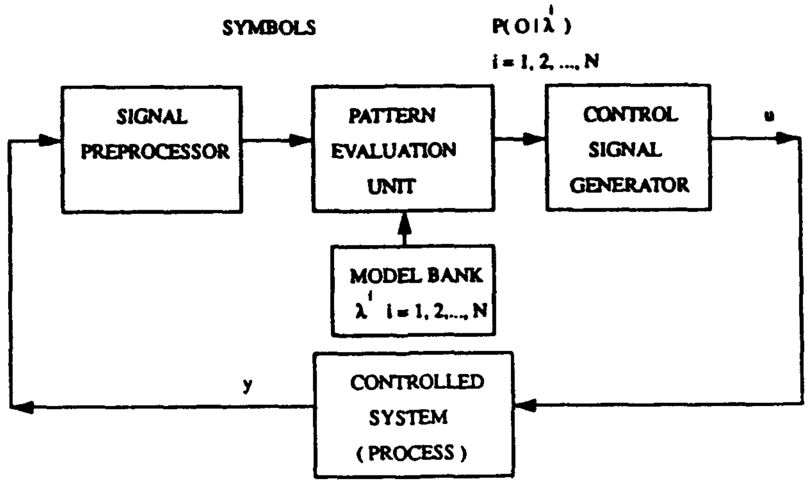


Figure 1: An HMM-based control system

The codebook is generated by the VQ algorithm. We use the LBG algorithm to produce the VQ codebook [20]. The LBG algorithm iteratively splits the training data into 2, 4, ..., 2^m partitions with a centroid for each partition.

For single-input single-output (SISO) systems, we can simply use a scalar quantization technique to map the feedback signal onto finite symbols.

3.2 Pattern Evaluation Unit

Pattern evaluation is the problem of determining the probability that a given HMM λ generates a sequence, $O = O_1 O_2 \cdots O_T$. The most straightforward way to calculate the probability of the observation sequence $O = O_1 O_2 \cdots O_T$ is through enumerating every possible state sequence of length T. For every fixed state sequence $S = S_1 S_2 \cdots S_T$, based on the assumptions we stated, the probability of the observation sequence O, $P(O|S,\ \lambda)$ can be computed as follows:

$$P(O|S,\ \lambda) = b_{s_1}(O_1) b_{s_2}(O_2) \cdots b_{s_T}(O_T). \tag{10}$$

On the other hand, the probability of a state sequence S can also be written as

$$P(S|\lambda) = \pi_{s_1} a_{s_1 s_2} a_{s_2 s_3} \cdots a_{s_{T-1} s_T}. \tag{11}$$

The joint probability of O and S, i.e., the probability that O and S occur simultaneously, is simply the product of the above two terms.

$$P(O,S|\lambda) = P(O|S,\lambda) P(S|\lambda). \tag{12}$$

The probability $P(O|\lambda)$ is the summation of this joint probability over all possible state

sequences:

$$
\begin{aligned}
P(O|\lambda) &= \sum_{all\ S} P(O|S,\lambda)P(S|\lambda) \\
&= \sum_{all\ S} \prod_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(O_t).
\end{aligned} \tag{13}
$$

Note that the computation for the above equation is on the order of $O(N^T)$. We must go through N possible states at every time $t = 1, 2, \cdots, T$. Even for small value of T and N, this computation is expensive. For example, if $N = 5$ and $T = 100$, the required computation is on the order of 10^{69}. To avoid this computation expense, we can use a more efficient algorithm known as the Forward-Backward algorithm [12].

Forward algorithm

The forward variable $\alpha_t(i)$ is defined as

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t,\ S_t = i|\lambda). \tag{14}$$

This is the probability of a partial observation sequence to time t, and state S_i which is reached at time t, given the model λ. This probability can be inductively computed by the following steps:

1. Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1) \quad 1 \leq i \leq N. \tag{15}$$

2. Induction:

$$
\begin{aligned}
\alpha_{t+1}(j) &= \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \\
&\quad 1 \leq t \leq T-1,\ \ 1 \leq j \leq N.
\end{aligned} \tag{16}
$$

3. Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{17}$$

With this algorithm, the computation of $\alpha_t(j)$ is only on the order of $O(N^2T)$.

Backward algorithm

In a similar way, the backward variable $\beta_t(i)$ is defined as

$$\beta_t(i) = P(O_{t+1} O_{t+2} \cdots O_T|S_t = i,\ \lambda). \tag{18}$$

i.e., the probability of a partial observation sequence from $t+1$ to the final observation T, given state i at time t and the model λ. The backward variable is computed in the following manner.

1. Initialization:

$$\beta_T(i) = 1, \quad 1 \leq i \leq N. \tag{19}$$

2. Induction:

$$\begin{aligned} \beta_t(i) &= \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \\ & \quad t = T-1, T-2, \cdots, 1, \ \ 1 \leq i \leq N. \end{aligned} \tag{20}$$

3. Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \pi_i b_i(O_1) \beta_1(i). \tag{21}$$

The computational complexity of $\beta_t(i)$ is approximately the same as that of $\alpha_t(i)$. Both the Forward and Backward algorithms can be used for computing $P(O|\lambda)$. However, in the problem of recognition, we need to computer $P(\lambda|O)$. Using Bayes' formula, $P(\lambda|O)$ can be obtained by

$$P(\lambda|O) = \frac{P(O|\lambda)P(\lambda)}{P(O)}. \tag{22}$$

3.3 Model Bank

In order to characterize the decision patterns, each pattern is described by an HMM. Since the feedback signal is a time sequence, the underlying state sequence associated with the model has the property that, as time increases, the state index increases (or stays the same), i.e., the states proceed from left to right. Supposing the control actions depend on the most recent m feedback samples, we can use an n ($n < m$) state left-right HMM, or so called Bakis model, to describe the pattern as shown in Figure 2.

The transition matrix in this case is

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & \cdots & 0 \\ 0 & a_{22} & a_{23} & a_{24} & 0 & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & a_{n-2,n-2} & a_{n-2,n-1} & a_{n-2,n} \\ \vdots & \ddots & \ddots & 0 & a_{n-1,n-1} & a_{n-1,n} \\ 0 & \cdots & \cdots & \cdots & 0 & a_{nn} \end{bmatrix}. \tag{23}$$

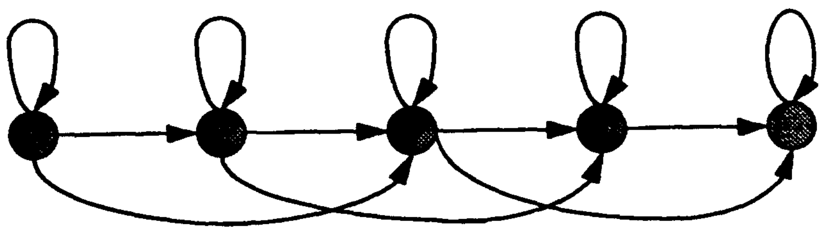


Figure 2: 5-state left-right HMM

Clearly this model has fewer parameters than that of ergodic, or fully connected HMMs. Furthermore, the initial state probabilities have the property

$$\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1. \end{cases} \tag{24}$$

Moreover, the state transition coefficients of state n are specified as

$$\begin{aligned} a_{nn} &= 1, \\ a_{n,i} &= 0, \quad i < n. \end{aligned} \tag{25}$$

If we convert the feedback signals into p symbols by certain signal processing techniques, B matrix is an $n \times p$ matrix.

Learning is achieved by adjusting the HMM parameters (A, B, π) to maximize the probability of the observation sequence. If the model parameters are known, we can compute the probabilities of an observation produced by given model parameters and then update the model parameters based on the current probabilities. if the model parameters are unknown, however, no analytic method is available for updating the model parameters.

An iterative algorithm is used to update the model parameters. For any model λ with non-zero parameters, we first define the posterior probability of transitions γ_{ij}, from state i to state j, given the model and the observation sequence,

$$\begin{aligned} \gamma_t(i,j) &= P(S_t = i,\ S_{t+1} = j | O, \lambda) \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}. \end{aligned} \tag{26}$$

Similarly, the posterior probability of being in state i at time t, $\gamma_t(i)$, given the observation sequence and model, is defined as

$$\begin{aligned} \gamma_t(i) &= P(S_t = i | O, \lambda) \\ &= \frac{\alpha_t(i)\beta_t(i)}{\sum_{k=1}^{N} \alpha_T(k)}. \end{aligned} \tag{27}$$

The expression $\sum_{t=1}^{T-1} \gamma_t(i)$ can be interpreted as the expected (over time) number of times that state S_i is visited, or, the expected number of transitions made from state S_i if time slot $t = T$ is excluded from the summation. Similarly, the summation of $\gamma_t(i,j)$ from $t = 1$ to $t = T - 1$ can be interpreted as the expected number of transitions from state S_i to state S_j.

Using the above formulas and the concept to count event occurrences, a new model $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$ can be created to improve the old model $\lambda = (A, B, \pi)$. A set of reasonable reestimation formulas for π, A, and B is:

$$\overline{\pi} = \gamma_1, \tag{28}$$

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i,j)}{\sum_{t=1}^{T-1} \sum_{j} \gamma_t(i,j)}, \tag{29}$$

$$\overline{b}_j(k) = \frac{\sum_{t \in O_t(i)=v_k} \gamma_t(j)}{\sum_{t}^{T} \gamma_t(j)}, \tag{30}$$

$$j = 1, 2, \ldots N,$$
$$k = 1, 2, \ldots M,$$

where v_k is the observation symbol.

Equations (28) to (30) are extensions of the Baum-Welch reestimation algorithm [22]. The Baum-Welch algorithm gives the maximum likelihood estimate of the HMM and can be used to obtain the model which describes the most likely performance for a given pattern. It has proved that either the initial model λ defines a critical point of the likelihood function, where new estimates equal old ones, or are more likely to produce the given observation sequence, i.e., the model $\overline{\lambda}$ is more likely than the model λ in the sense that $P(O|\overline{\lambda}) \geq P(O|\lambda)$.

By repeating the above reestimation and replacing λ with $\overline{\lambda}$, we ensure that $P(O|\lambda)$ can be improved until a limiting point is reached.

3.4 Control Signal Generator

Equation (22) provides a way to evaluate how well the feedback signal matches the HMMs in the model bank. Because $P(O)$ is a constant for a given input, only $P(O|\lambda)P(\lambda)$ needs computing. $P(\lambda)$ is the probability that the control pattern is used. This problem can be solved with a syntactic method [21].

Syntactic methods assume that patterns are composed of subpatterns in the same way that phrases and sentences are built by concatenating words and words are built by concatenating

characters. The structure information of the patterns is provided by the so-called "pattern description language." The "grammar" of the pattern description language governs the composition of subpatterns into patterns. Once each subpatterns within the pattern has been recognized, the recognition process is accomplished by performing a syntax analysis. The syntax analysis generates a structure description of the sentence representing the given pattern. Based on the syntactic approach, $P(\lambda)$ is determined by the "grammar.' The grammar determines the probability with which one control action is followed by another action. This structure information can introduce control history into current control output to avoid unexpected control output. $P(\lambda)$ can be learned from the training data or assigned by experience. In the simplest case, if all the control patterns are equally likely to be used, only the term $P(O|\lambda)$ is a variable. In this case, Equation (8) can be rewritten as:

$$u = \frac{\sum_{j=1}^{m} P(O|\lambda^j) \cdot u_j}{\sum_{j=1}^{m} P(O|\lambda^j)}. \tag{31}$$

Since only a single value $u(j)$, $j = 1, 2, \ldots, m$, is used for each pattern, there is a need to compute this single value. One way to do this is to simply take the center point of the pattern in the control space. An alternative method is to use mean value of all the training data belonged to the pattern. We have tried both methods and found that both of them were effective.

4 Case Studies

To evaluate the validity and effectiveness of the proposed method, we have carried out two case studies. we first examined HMM-based controller for a sun-seeker control system. The system is mounted on a space vehicle to track the sun. We then applied the HMM learning controller to a benchmark problem in intelligent control – inverted pendulum balancing. The problem is challenging because of its nonlinear unstable behavior.

Conceptually, if we can obtain good results with these systems, we have sufficient reasons to expect the HMM controller to work well for other systems, because the only difference lies in the data measurement and processing. We performed simulations to examine the feasibility of the method for handling well-defined problems and to reveal various important issues such as parameterization and stability.

4.1 Linear System

The system is shown in Figure 3, and the transfer function of the plant is:

$$G_p(s) = \frac{2500}{s(s+25)}. \tag{32}$$

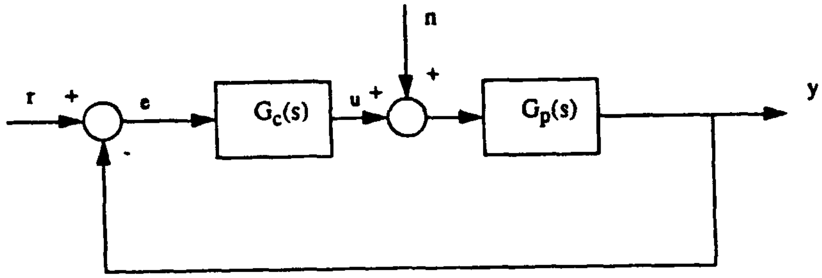


Figure 3: Block diagram of the sun-seeker control system

This is a second order linear system. To obtain the training data, we employed a pł se-lead controller as the "teacher." The controller transfer function is:

$$G_c(s) = \frac{1 + 0.0342s}{1 + 0.00588s}. \tag{33}$$

The noise $n(t)$ is the Gaussian white noise with zero mean and variance 0.01, i.e.,

$$E\{n(t)\} = 0, \quad E\{n(t)n(t)^T\} = 0.01. \tag{34}$$

The input/output data of the controller were recorded while the phase-lead controller controlled the system. We assumed that the decision patterns are related to ten consistent samples of error signals. This means that ten samples of error signals are used for decision making. Because this is an SISO system, we can use a scalar quantization technique to convert the error signal into finite symbols. The feedback signal was quantized into 256 levels and the controller output was based on 26 decision patterns. Therefore, there is a total of 26 HMMs in the model bank and 256 symbols in the output probability distribution functions of each discrete HMM. Each input sampling corresponded to one of 256 symbols and each of 10 symbols is associated with an HMM. To describe the patterns, we used five-state left-right HMMs. Let $n = 5$ we can obtain the form of transition matrix A, the initial state probabilities, and the state transition coefficients of state 5 from equations (24) to (25). The observability matrix B is a 256×5 matrix where each column represents the observation probability distribution for one state.

The training data were obtained by providing a unit square wave with a changeable width as reference input to the system (see Figure 3). After collecting 9000 samples of controller input/output data, we partitioned each control command into one of the 26 patterns and converted each feedback sampling into one of 256 symbols. Then we handmarked each of 26 patterns with corresponding feedback symbols for training. To initialize the model parameters, we set output probabilities to $\frac{1}{256}$, where 256 is the quantization level. The transition probabilities were initialized by the uniformly distributed random number. With these initial parameters, the Forward-Backward algorithm was run recursively on the training data.

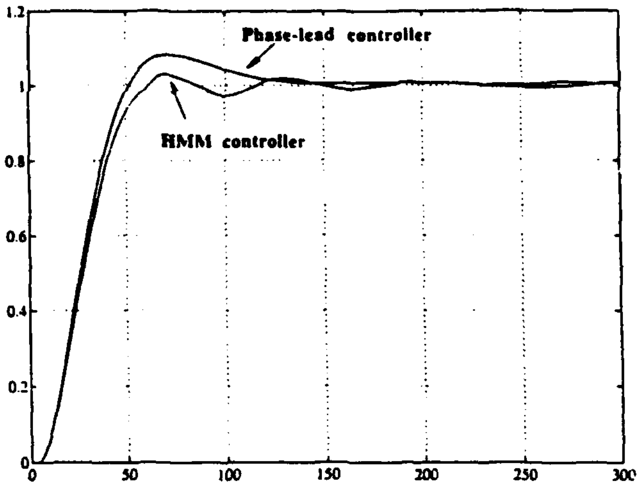


Figure 4: Step responses of the system

The Baum-Welch algorithm was used iteratively to reestimate the parameters based on the forward and backward variables. After each iteration, the output probability distributions were smoothed using a floor between 0.0001 and 0.00001 and renormalized to meet stochastic constraints. Twelve iterations were run for the training processes.

The trained HMMs were then put into the model bank and the HMM-based controller was employed to control the system. Figure 4 shows a comparison of step responses between the phase-lead controlled system and the HMM-based control system. We notice that the overshoot of the HMM-based system is less than that of the phase-lead controller system, but the steady state error of the HMM-based system is larger. Figure 5 shows tracking performance of the HMM-based controller. The HMM-based control system tracks a square wave reference input well. In all simulations, we added the Gaussian white noise with zero mean and variance 0.01. These results demonstrate that HMMs are able to learn a control strategy and to control systems.

In the HMM-based controller, the probability of an HMM matching with the feedback signal pattern determines the contribution of the HMM to controller output, i.e., the probability is a weight. However, there is no criterion for selecting m in Equation(8). The relation between m and summation of square error for the system discussed in this report is shown in Figure 6. The summation of square error is a minimum when $m = 8$ and remains unchanged when $m \geq 13$. This implies that for computing control outputs, a large number of the patterns does not guarantee better performance.

4.2 Inverted Pendulum

The inverted pendulum system shown in Figure 7 consists of an inverted pendulum of length L and mass m and a cart of mass M. The pivot of the pendulum is mounted on a cart

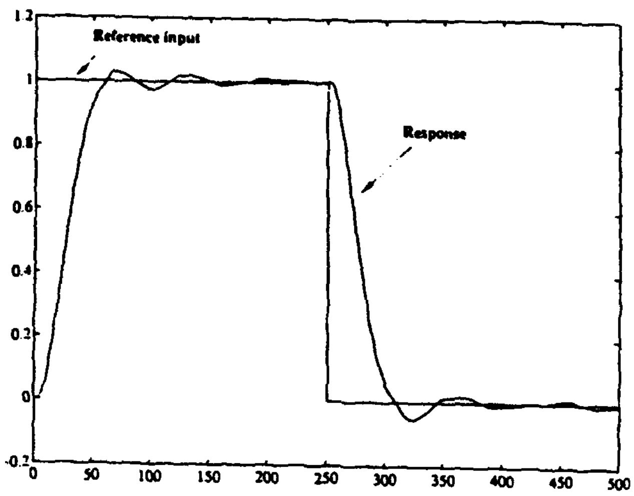


Figure 5: Response of HMM-based control system to a square wave input

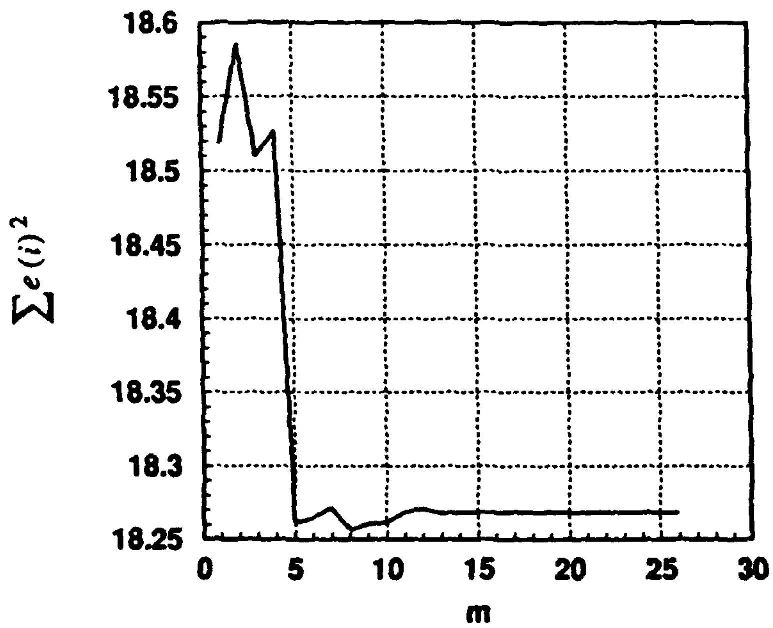


Figure 6: The relation between m and $\sum_{i=1}^{m} e(i)$

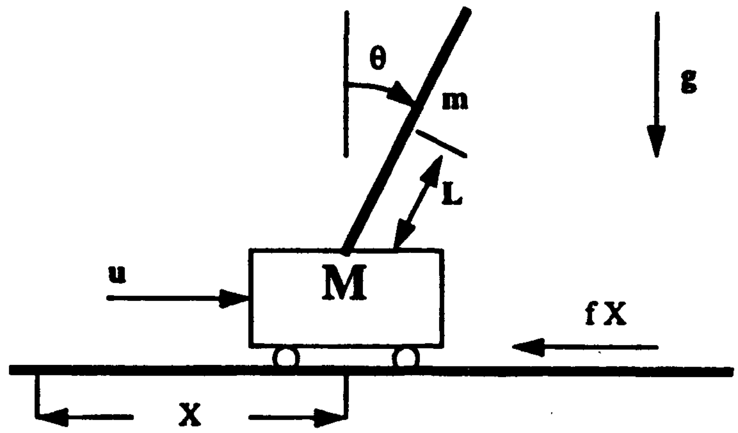


Figure 7: Inverted pendulum system

which can move in a horizontal direction. The cart is controlled by the horizontal force u, the input to the system.

We assume that the pendulum is a narrow, uniform rod and there is no friction at the pivot and no actuator dynamics. The inverted pendulum system was simulated using the following equations of motion:

$$\ddot{\theta} = \frac{3}{4L}(g \sin\theta - \ddot{X}\cos\theta), \tag{35}$$

$$\ddot{X} = \frac{m(L\sin\theta\dot{\theta}^2 - \frac{3}{8}g\sin 2\theta) - f\dot{X} + u}{M + m(1 + \frac{3}{4}\cos^2\theta)}, \tag{36}$$

where $M = 1$ (kg), $m = 0.1$ (kg), $L = 1$ (m), $f = 5$ (kg/s), and $g = 9.81$ (m/s^2).

This system was simulated numerically using Euler's approximation method, $\theta[k+1] = \theta[k] + T\dot{\theta}[k]$, with a time step $T = 0.02$ second. The sampling rate of the inverted pendulum's states and the rate at which control forces are applied are the same as the simulation rate, i.e., 50 Hz. Our control goal here was to balance the pendulum by applying a sequence of right and left forces regardless of the cart's position and velocity. We assumed the pendulum angle θ to be measurable. To obtain the training data, we employed nonlinear control law [23]:

$$h_1 = \frac{3}{4L}g\sin\theta, \tag{37}$$

$$h_2 = \frac{3}{4L}\cos\theta, \tag{38}$$

$$f_1 = m(L\sin\theta\dot{\theta}^2 - \frac{3}{8}g\sin 2\theta) - f\dot{X}, \tag{39}$$

$$f_2 = M + m(1 - \frac{3}{4}\cos^2\theta), \tag{40}$$

$$u = \frac{f_2}{h_2}\left[h_1 + k_1(\theta - \theta_d + k_2\dot{\theta}\right] - f_1, \tag{41}$$

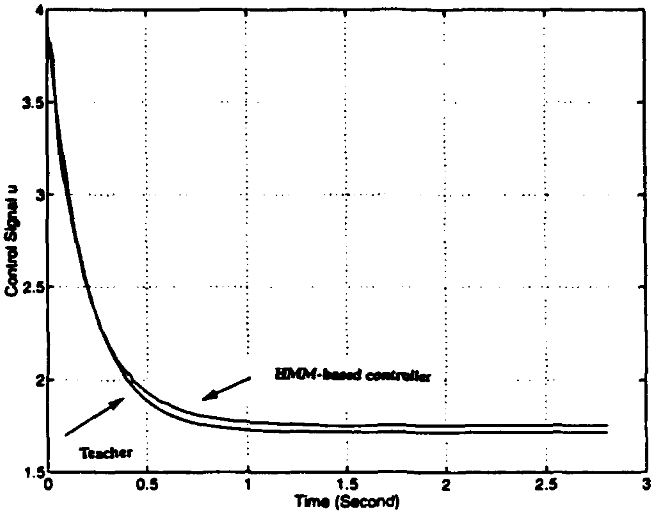
where $k_1 = 25$, $k_2 = 10$.

In order to obtain training data, 80 iterations of simulation were conducted with the above control law and initial state $[X, \dot{X}, \theta, \dot{\theta}]^T = [1.0, 0.0, 5.0, 0.0]^T$. The noise with zero mean and 0.01 variance was added to initial value of θ to avoid the same data for all iterations. The values of θ and u were recorded for training the HMM. The feedback signal θ was quantized into 256 levels and partitioned into 40 decision patterns. Therefore, a total of 40 HMMs, each of which contains 6 states, was employed to encode decision patterns. Each HMM was trained by corresponding data. Eight sequential samples of feedback signal were evaluated each time to determine control patterns. Figure 8 compares the response of the HMM-based control system and the ideal response of the nonlinear control system.

5 Conclusion

In many real world practices, the control strategy cannot be explicitly expressed as a function, but can be given by examples. Learning from examples is a feasible way to develop a controller automatically. If the control strategy has uncertainties, the learning method must have the mechanism to cope with stochastic nature. HMM is a powerful parametric model and is feasible to characterize a doubly stochastic process. This report presents a learning controller based on HMM to characterize and learn the decision patterns. We have developed a technique to build an HMM-based learning controller. We demonstrated the feasibility of the proposed method by simulation studies. We investigated two cases: a linear system and an inverted pendulum system.

Although we have examined only cases of well known controllers as teachers, the teacher can be any type controller, including control strategies used by a human. For a MIMO system, it is possible to use a multi-dimensional HMM to encode the control strategy. A multi-dimensional HMM is an HMM which has more than one observable symbol at each time t and is capable of dealing with multiple feedback signals. A multi-dimensional HMM is also appropriate for fusing different sensory signals with different physical meanings. However, to use HMMs for an MIMO system, more efficient method must be developed to generate multiple control outputs. A possible way is to use the VQ technique for partitioning the control space and then to use the same method as an SISO system for generating each component of control vector. The future research on HMM-based strategy modeling lies in both theory and practice. Because a HMM-based controller is nonlinear, it is difficult to analyze the HMM-based system. A theory is needed to analyze the stability and performance of the system. In practice, more realistic problems should be investigated which leads to the development of new methods for designing HMM-based controller. One short term extension is to develop a systematic procedure for optimally partitioning control space.



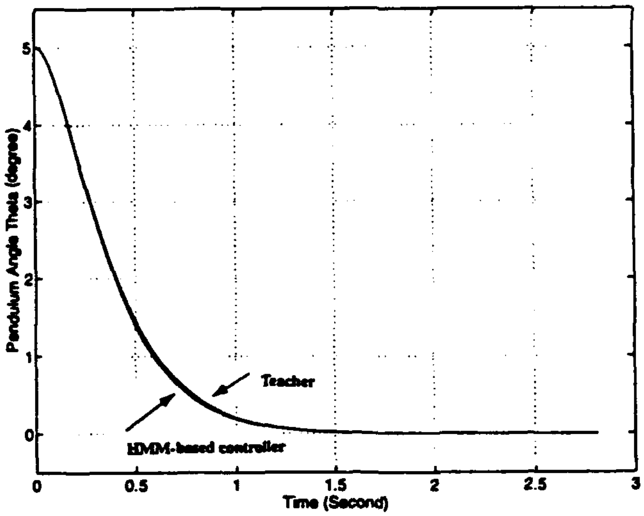


Figure 8: Responses of HMM-based system and the teacher

Acknowledgement

The authors would like to thank Dr. T. Kanade, and Dr. C.S. Chen for their valuable suggestions and discussions.

References

[1] G. Saridis, "Application of pattern recognition methods to control systems," *IEEE Trans. on Automatic Control*, Vol. AC-26, No.3, pp. 638-645, 1981.

[2] K.S. Fu, "The theoretical background of recognition as applicable to industrial control," *Learning System and Pattern Recognition in Industrial Control (Proceedings of the Ninth Annual Advanced Control Conference)*, pp. 29-40, 1983.

[3] C.C. Lee, "Fuzzy logic in control systems: fuzzy logic controller – part I," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 20, No. 2, pp. 404-418, 1990.

[4] C.C. Lee, "Fuzzy logic in control systems: fuzzy logic controller, part II," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 20, No. 2, pp. 419-435, 1990.

[5] Panos J. Antsaklis, Guest Editor, Special issue on neural networks in control systems, *IEEE Control System Magazine*, Vol.10, No. 3, pp. 3-87, 1990.

[6] W.T. Miller, R. S. Sutton and P.I. Werbos, Editors, " Neural network for control," *MIT press*, 1990.

[7] Panos J. Antsaklis, Guest Editor, Special issue on neural networks in control systems, *IEEE Control System Magazine*, Vol.12, No. 2, pp. 8-57, 1992.

[8] B. Kosko, "Neural Network and Fuzzy Systems," Englewood Cliffs, NJ, Prentice Hall, 1992.

[9] Panos J. Antsaklis, "Neural network in control systems," *IEEE Control System Magazine*, Vol.12, No. 2, pp. 8-10, 1992.

[10] L.R. Rabiner, "A Tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, Vol. 77, No. 2, pp.257-286, 1989.

[11] K.F. Lee, H.W. Hon and R. Reddy, "An overview of the SPHINX speech recognition system," *IEEE Trans. on ASSP*, Vol. 38, No. 1, pp. 35-45, 1990.

[12] X.D. Huang, Y. Ariki and M. A. Jack, "Hidden Markov Models for Speech Recognition," *Edinburgh University Press*, 1990.

[13] X.D. Huang, "Phoneme classification using semicontinuous hidden Markov models," *IEEE Trans. on ASSP*, Vol. 40, No. 5, pp. 1062-1067, 1992.

[14] B. Hannaford, P. Lee, "Hidden Markov model analysis of force/ torque information in telemanipulation," *The International Journal of Robotics Research*, Vol. 10, No. 5, pp.528-539, 1991.

[15] P. K. Pook and D. Ballard, "Recognizing teleoperated manipulations," *Proceedings of 1993 IEEE Inter. Conf. on Robotics and Automation*, Atlanta, Georgia, USA, Vol. 2, pp.578-585, 1993.

[16] V. Krishnamurthy, J.B. Moore, and S.-H. Chung, "Hidden Markov model signal processing in presence of unknown deterministic interferences," *IEEE Trans. Autom. Control*, Vol.38, No.1, pp. 146-152, 1993.

[17] J. Yang, Y. Xu and C.S. Chen, "Hidden Markov model approach to skill learning and its application in telerobotics," *Proceedings of 1993 IEEE International Conference on Robotics and Automation*, pp. 396-402, 1993.

[18] J. Yang, Y. Xu, and C.S. Chen, *Gesture interface: modeling and learning*, *Proceedings of 1994 IEEE International Conference on Robotics and Automation*, May 8-13, 1994.

[19] R.M. Gray, "Vector quantization," *IEEE ASSP Magazine*, Vol. 1, No.2, pp. 4-29, 1984.

[20] Y. Linde, A. Buzo, and R.M. Gray, " An algorithm for vector quantizer design," *IEEE Trans. on Commun.*, Vol. COM-28 pp.84-95, 1980.

[21] K. S. Fu, "Syntactic Methods in Pattern Recognition," *Volume 112 of Mathematics in Science and Engineering*, New York: Academic Press, 1974.

[22] L.E. Baum, T. Petrie, G. Soules, and N.Weiss, " A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chins," *Ann. Math. Stat.*, Vol. 41, No. 1, pp. 164-171, 1970.

[23] A. Guez and J. Selinsky, "A trainable neuromorphic controller," *Journal of Robotic Systems*, Vol.5, No.4, pp. 363-388, 1988.